Preprint version Workshop on Robot Motion Planning: Online, Reactive, and in Real-time, 2012 IEEE/RSJ IROS, Vilamoura, Portugal

A Synergetic High-level/Reactive Planning Framework with Application to Human-Assisted Navigation

Antonio Franchi, Carlo Masone and Paolo Robuffo Giordano

***Abstract*— In this work we present a novel framework for the systematic integration of high-level/mission schedulers, middle-level/cognitive-enabled online-planners and low-level/reactive trajectory modifiers. The approach does not rely on a particular parametrization of the trajectory and assumes a basic environment representation. As an application, the online capabilities of the method can be used to let a mobile robot cooperate with a human taking the role of the middle-level planner. In that case we also describe a rigorous way to bilaterally couple the human and the reactive planner in order to provide an immersive haptic feeling of the planner state. Hardware/Human in-the-loop simulations, with a quadrotor UAV used as robotic platform and a real haptic instrument, are provided as validating showcase of the presented theoretical framework.**

I. Introduction

Combination of low-level controllers with high-level/sophisticated planning abilities represents one of the crucial issues to enable complex decision making in real-world unstructured scenarios. Furthermore, the presence of a systematic framework to combine these two aspects may also result helpful in many task requiring human-robot interaction and cooperation, e.g., in order to optimally balance the human commitments and robot autonomy.

In these notes we focus on the very common scenario where a mobile robot is tasked to navigate in an environment in order to accomplish some given mission, e.g., exploration, surveillance, monitoring, search-and-rescue, good transportation, mobile-networking, etc.. Some prior knowledge on the environment may be given (e.g., the environment size and a rough map that has been retrieved with a preliminary exploration). The environment is also populated with obstacles and points-of-interests that can only be detected when the robot is sufficiently close (e.g., victims, goods to be loaded, charging docks, stationary antennas, etc.). The planning step is then divided in three phases: mission scheduling, online middle-level planning, and reactive trajectory modifier. Our deformation differs from other well known approaches that apply artificial forces to a sequence of configurations [1], [2], or define trajectory modifications as input functions along the admissible directions of motion [3], because it does not arbitrarily change the path but only some desired geometric properties.

We also propose a systematic way to incorporate a human assistant as online middle-level planner, in order to exploit his/her advanced cognitive capabilities. In the case of a human in-the-loop we propose a haptic algorithm that feeds back to the human a force cue informative of the *global* deformation acting on the desired path rather than on a *local* mismatch between commanded and executed position/velocity, as in all the previous works in our knowledge, see, e.g., [4], [5], [6], [7], [8].

Summarizing, the main contributions of our framework are: i) a systematic integration of three different motion planning layers that does not rely on a particular trajectory parametrization and uses a basic representation of the surrounding environment ii) the possibility of seamlessly applying this framework for including, online and in real-time, a human operator in the planning loop in order to exploit her/his superior cognitive skills, iii) the design of a general force-cue paradigm that closes the loop between the automatic part of the motion planner and the human assistant (when present), and iv) the fact the the proposed force cue is informative of the global deformation of the desired path rather than of the mismatch between direct motion commands and their execution, as in previous bilateral teleoperation frameworks.

This framework is the generalization of the ideas presented in [9] where only the case of human operators and persistent trajectories are considered. In these notes we also present additiona plots from new simulation results.

II. Synergetic Trajectory Planning

The proposed planning framework is constituted by three main sub-systems: i) a high-level *task scheduler*, ii) a *middle-level modifier*, iii) a *reactive modifier*, as depicted in Fig. 1. In the case that the middle-level modifier duties are performed by a human assistant (as described in Sec. III) then an additional sub-system is represented by the *bilateral-controller* that provides the assistant with a suitable haptic feedback.

The three fundamental blocks operate in cascade. The task scheduler (TS) periodically generates an *initial reference path* that is intended valid for a given time horizon and it is based on past information. For example, it can be an exploration algorithm that plans the next move based on the current partial map or a coverage method that switches between predefined curve patterns. The middle-level modifier (MM) builds upon the initial reference path in order to generate online a time-varying *reference trajectory*. The underlying idea is that the MM comprises a sophisticated algorithm (or even a human assistant) that is able to promptly take into account new pieces of information gathered by the mobile robot and refine in real-time the reference motion by resorting on some sophisticated/cognitive capabilities. Finally the reactive modifier (RM) goal is to generate online

A. Franchi, C. Masone, and P. Robuffo Giordano are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {antonio.franchi,carlo.masone, prg}@tuebingen.mpg.de.

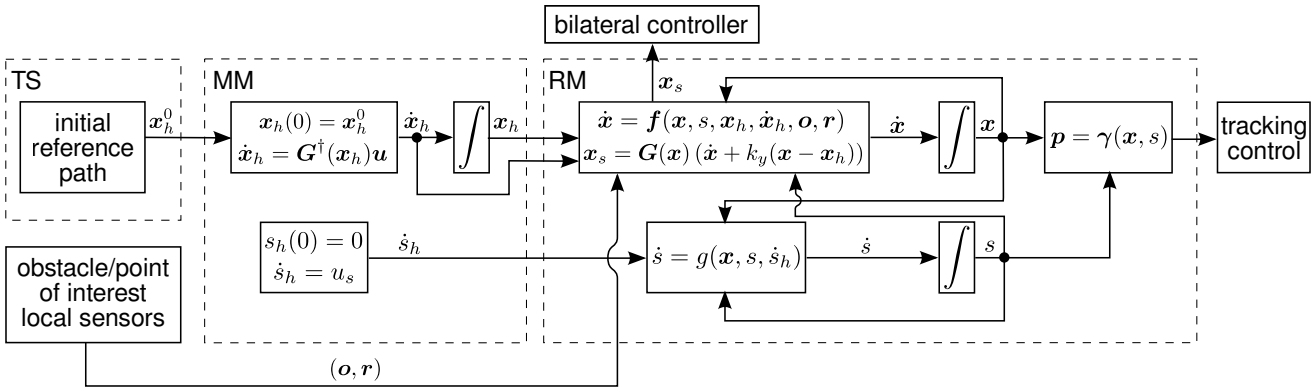


Fig. 1: Block diagram of the planning and control framework, with highlighted the high-level task scheduler (TS), the middle-level modifier (MM), and the low-level reactive modifier (RM).

the actually *tracked trajectory* for the robot motion controller in order to meet the following specifications:

1) ensuring feasibility of the motion, given the robot kinematic and dynamic constraints,
2) let the tracked trajectory be as much as possible similar to the reference trajectory,
3) ensure obstacle avoidance,
4) pass close to the points-of-interest that are located in the vicinity of the reference trajectory.

The reference trajectory, generated online by the MM, is specified as a *geometric path*

$$\boldsymbol{\gamma} : \mathbb{R}^n \times [0, L_h] \to \mathbb{R}^d, \quad \text{s.t.} \quad (\boldsymbol{x}_h, s_h) \mapsto \boldsymbol{\gamma}(\boldsymbol{x}_h, s_h)$$

(where $\boldsymbol{x}_h \in \mathbb{R}^n$ is a vector of shape parameters uniquely defining the curve, L_h is the curve length, and s_h is the arc length of the curve) together with a *timing law* $s_h(t)$ which ultimately determines how the robot should travel the path.

At a certain initial time t_0 the TS provides the initial reference path to the MM in the form of an initial set of geometric parameters $\boldsymbol{x}_h^0$ for the reference trajectory, Then the MM sets $\boldsymbol{x}_h(t_0) = \boldsymbol{x}_h^0$ and $s_h = 0$ and from that moment the MM is free to modify the reference trajectory by acting on $\dot{\boldsymbol{x}}_h$ and $\dot{s}_h$. The current reference trajectory stays alive until the TS provides a new initial reference path to the MM. This event brings to a reset of the reference trajectory to the new initial reference path, and so on.

The number of parameters n and their kind depends on the specific representation used for $\boldsymbol{\gamma}$ and in general a larger n results in a higher flexibility of the geometric path. Nevertheless, managing the total number n of parameters required to cope with a typical unstructured environment may demand a too high computational load on the cognitive-planning side (e.g., it may exceed the number of quantities that a human operator can reasonably control at once).

For this reason we consider a vector $\boldsymbol{y}(\boldsymbol{x}_h) = (y_1(\boldsymbol{x}_h) \dots y_m(\boldsymbol{x}_h))^T \in \mathbb{R}^m$, $m \leq n$, defining the m degrees of freedom assigned to the middle-level modifier. The time variation of $\boldsymbol{y}$ is given by

$$\dot{\boldsymbol{y}} = \left(\frac{\partial y_1}{\partial \boldsymbol{x}_h}^T \cdots \frac{\partial y_m}{\partial \boldsymbol{x}_h}^T \right)^T \dot{\boldsymbol{x}}_h = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}_h} \dot{\boldsymbol{x}}_h = \boldsymbol{G}(\boldsymbol{x}_h) \dot{\boldsymbol{x}}_h, \tag{1}$$

where matrix $\boldsymbol{G}(\boldsymbol{x}_h) \in \mathbb{R}^{m \times n}$ is assumed to have full row-rank so that the m dofs controlled by the MM are independent.

The action of the MM on the reference trajectory is then obtained by means of the following dynamical system

$$\dot{s}_h = u_s \tag{2a}$$
$$\dot{\boldsymbol{x}}_h = \boldsymbol{G}^\dagger(\boldsymbol{x}_h)\, \boldsymbol{u}, \tag{2b}$$

where $u_s \in \mathbb{R}$ and $\boldsymbol{u} \in \mathbb{R}^m$ represent the actual MM commands (whose detailed expression in the case of a human operator is described in Sec. III), and $\boldsymbol{G}^\dagger$ is the pseudo-inverse of matrix $\boldsymbol{G}$.

A. Pure-reactive Dynamics of the Tracked Trajectory

The environment is considered populated by a set of obstacles, described by the vector of obstacle points $\boldsymbol{o} = (\boldsymbol{o}_1 \dots \boldsymbol{o}_{n_o}) \in \mathbb{R}^{d \times n_o}$, and a set of regions of interest, described by the vector of points of interest $\boldsymbol{r} = (\boldsymbol{r}_1 \dots \boldsymbol{r}_{n_r}) \in \mathbb{R}^{d \times n_r}$. Given the reference trajectory parameters $(s_h, \boldsymbol{x}_h)$ and the environment $(\boldsymbol{o}, \boldsymbol{r})$, the reactive modifier generates the tracked trajectory $\boldsymbol{p}(t)$ as dictated by the following dynamical system:

$$\dot{s} = g(\boldsymbol{x}, s, \dot{s}_h) \tag{3a}$$
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, s, \boldsymbol{x}_h, \dot{\boldsymbol{x}}_h, \boldsymbol{o}, \boldsymbol{r}) \tag{3b}$$
$$\boldsymbol{p} = \boldsymbol{\gamma}(\boldsymbol{x}, s), \tag{3c}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $s \in [0, L]$ are the shape parameters and the arc length of the tracked trajectory, respectively, and the initial conditions are of the system as set as $\boldsymbol{x}(t_0) = \boldsymbol{x}_h(t_0) = \boldsymbol{x}_h^0$, and $s(t_0) = 0$.

The arc length map (3a) has the following form

$$g(\boldsymbol{x}, s, \dot{s}_h) = g_1(\boldsymbol{x}, s, \dot{s}_h)\dot{s}_h, \tag{4}$$

where $g_1 \in [0, 1]$ is designed so as to cope with the robot motion capabilities. In particular, $g_1 = 1$ if the robot can travel along the path at the desired speed $\dot{s}_h$, and $g_1 \to 0$ (thus, towards a full stop) whenever the speed $\dot{s}_h$ becomes too large for the actuation capabilities of the robot (e.g., because of a too large curvature or a low energy level for the robot batteries).

Concerning the vector field $\boldsymbol{f}$, we first show how to design it in order to meet the first specification, i.e., feasibility. In particular we want to prevent that at any time the geometric properties of $\boldsymbol{\gamma}(\boldsymbol{x},\, s)$ are directly influenced by the variation $\dot{\boldsymbol{x}}$ given by (3b) This is important to ease the action of the robot trajectory tracker which expects presence of some local regularity properties of the curve being tracked[1].

Local geometric properties of a curve are characterized by the k-th derivatives w.r.t. the arc length s, i.e.,

$$\boldsymbol{p}^{(k)}(\boldsymbol{x}, s) = \boldsymbol{\gamma}^{(k)}(\boldsymbol{x}, s) = \frac{\partial^k \boldsymbol{\gamma}}{\partial s^k}(\boldsymbol{x}, s), \quad k \in \mathbb{N}_0$$

where the operator $\cdot^{(k)}$ indicates the k-th geometric derivative. By applying the chain rule, the variation of $\boldsymbol{p}^{(k)}$ due to changes of $\boldsymbol{x}$ and s over time is given by

$$\frac{d}{dt}\left(\boldsymbol{p}^{(k)}(\boldsymbol{x}, s)\right) = \left.\frac{\partial \boldsymbol{\gamma}^{(k)}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x},s)} \dot{\boldsymbol{x}} + \left.\frac{\partial \boldsymbol{\gamma}^{(k)}}{\partial s}\right|_{(\boldsymbol{x},s)} \dot{s}. \quad (5)$$

Stacking eq. (5) for the first k derivatives then yields

$$\dot{\boldsymbol{\Gamma}}_{0,k}(\boldsymbol{x}, s) = \boldsymbol{J}(\boldsymbol{x}, s)\dot{\boldsymbol{x}} + \boldsymbol{\Gamma}_{1,k+1}(\boldsymbol{x}, s)\dot{s}. \quad (6)$$

where $\boldsymbol{\Gamma}_{i,j} = [\boldsymbol{\gamma}^{(i)^T} \dots \boldsymbol{\gamma}^{(j)^T}]^T \in \mathbb{R}^{2(j-i+1)}$ and $\boldsymbol{J} \in \mathbb{R}^{2(k+1)\times n}$. Equation (6) shows how the geometric properties of the curve γ at some $(\boldsymbol{x}, s)$ depend on two contributions: the first is due to the parameter change $\dot{\boldsymbol{x}}$ and the second to the longitudinal speed $\dot{s}$. The feasibility requirement on $\boldsymbol{f}$ can then be interpreted as imposing that the first term in (6) is zero when evaluated at the current $(\boldsymbol{x},\, s)$. Therefore, the design of the vector field $\boldsymbol{f}$ must ensure that

$$\boldsymbol{J}(\boldsymbol{x}(t), s(t))\dot{\boldsymbol{x}} = \boldsymbol{0}. \quad (7)$$

Assuming matrix $\boldsymbol{J}$ has a non-empy null-space, an infinity of possible $\dot{\boldsymbol{x}} \in \mathcal{N}(\boldsymbol{J})$ would meet the constraint. To this end, let $\boldsymbol{J}^{\dagger}(\boldsymbol{x}, s)$ represent the pseudo-inverse of $\boldsymbol{J}(\boldsymbol{x}, s)$, and $\boldsymbol{N}(\boldsymbol{x}, s) \in \mathbb{R}^{n\times n}$ a projector matrix spanning the null space $\mathcal{N}(\boldsymbol{J})$, i.e., such that $\boldsymbol{J}\boldsymbol{N} = \boldsymbol{0}$. We then design the vector field $\boldsymbol{f}$ to have the following form[2]

$$\boldsymbol{f}\left(\boldsymbol{x}, s, \boldsymbol{x}_h, \dot{\boldsymbol{x}}_h, \boldsymbol{o}, \boldsymbol{r}\right) = \boldsymbol{N}(\boldsymbol{x}, s)\boldsymbol{f}_1\left(\boldsymbol{x}, \boldsymbol{x}_h, \dot{\boldsymbol{x}}_h, \boldsymbol{o}, \boldsymbol{r}\right). \quad (8)$$

Assuming that matrix $\boldsymbol{J}$ has full row-rank, a well-known choice for the projector operator is $\boldsymbol{N} = (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})$.

In order to meet the remaining three specifications, we then design $\boldsymbol{f}_1$ in (8) as the composition of four terms:

$$\boldsymbol{f}_1 = \boldsymbol{f}_h(\boldsymbol{x}, \boldsymbol{x}_h, \dot{\boldsymbol{x}}_h) + \boldsymbol{f}_o(\boldsymbol{x}, \boldsymbol{o}) + \boldsymbol{f}_r(\boldsymbol{x}, \boldsymbol{r}) + \boldsymbol{f}_i(\boldsymbol{x}). \quad (9)$$

The first term implements a feedforward/proportional action

$$\boldsymbol{f}_h(\boldsymbol{x}, \boldsymbol{x}_h, \dot{\boldsymbol{x}}_h) = \dot{\boldsymbol{x}}_h + k_h(\boldsymbol{x}_h - \boldsymbol{x}), \quad (10)$$

with $k_h > 0$, in order to steer $\boldsymbol{x}$ towards the reference $\boldsymbol{x}_h$; $\boldsymbol{f}_o(\boldsymbol{x}, \boldsymbol{o})$ is a vector field that moves $\boldsymbol{\gamma}$ away from the obstacles $\boldsymbol{o}$; $\boldsymbol{f}_r(\boldsymbol{x}, \boldsymbol{r})$ is a vector field that attracts $\boldsymbol{\gamma}$ towards the points of interest $\boldsymbol{r}$; and $\boldsymbol{f}_i(\boldsymbol{x})$ is an additional vector field that is left free to the engineer in order to implement some additional internal properties of the curve that may be of interest, e.g., internal elasticity, stiffness, or viscosity.

With regard to $\boldsymbol{f}_o$, each obstacle $\boldsymbol{o}_i$, $i = 1, \dots, n_o$, implements a strictly monotonic and scalar potential $\varphi_{oi} : \mathbb{R}_0 \to \mathbb{R}^+$ such that, for a generic point $\boldsymbol{\gamma}(\boldsymbol{x}, s)$ on the path, $\varphi_{oi} \to \infty$ when $\|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{o}_i\| \to 0$, $\varphi_{oi} \to 0$ smoothly when $\|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{o}_i\| \to R_{oi}$ and $\varphi_{oi} \equiv 0$ when $R_{oi} > 0$ where $R_{oi} > 0$ defines the region of influence of the obstacle. The action of the anti-gradient vector field of φ_{oi} on the point $\boldsymbol{\gamma}(\boldsymbol{x}, s)$ is

$$\boldsymbol{f}^{\boldsymbol{p}}_{oi}(\boldsymbol{x}, s, \boldsymbol{o}_i) = -\frac{\partial \varphi(\|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{o}_i\|)}{\partial \boldsymbol{\gamma}(\boldsymbol{x}, s)}. \quad (11)$$

The overall action exerted on the curve by a single obstacle and projected on the shape parameters space is

$$\boldsymbol{f}_{oi}(\boldsymbol{x}, \boldsymbol{o}_i) = \int_{\boldsymbol{\gamma}} \left.\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}}\right|^{\dagger}_{(\boldsymbol{x},s)} \boldsymbol{f}^{\boldsymbol{p}}_{oi}(\boldsymbol{x}, s, \boldsymbol{o}_i) ds, \quad (12)$$

where the previous considerations on the existence of the pseudo-inverse still hold. Finally, the total action of the obstacles is just the sum over all the elements in $\boldsymbol{o}$:

$$\boldsymbol{f}_o = \sum_{i=1}^{n_o} \boldsymbol{f}_{oi}(\boldsymbol{x}, \boldsymbol{o}_i), \quad (13)$$

We note that, from a practical standpoint, the analytical expression of (11) can be hard to determine, so that a numerical evaluation of the integral may be needed.

By resorting to similar arguments, we define an attractive vector field $\boldsymbol{f}_{ri}(\boldsymbol{x}, \boldsymbol{r}_i)$ for each point of interest $\boldsymbol{r}_i$, and sum each contribution in order to obtain

$$\boldsymbol{f}_r = \sum_{i=1}^{n_r} \boldsymbol{f}_{ri}(\boldsymbol{x}, \boldsymbol{r}_i). \quad (14)$$

B. Obstacle-Crossing Dynamics for the Tracked Trajectory

One drawback of using artificial potentials whose intensity becomes infinite as the distance to the obstacles goes to zero is that it does not allow the tracked trajectory to cross over an obstacle even if this could result in a smaller error norm $e(\boldsymbol{x}, \boldsymbol{x}_h) = \|\boldsymbol{x}_h - \boldsymbol{x}\|$. This limitation is a well known problem in the reactive planning literature and it has been tackled in different ways. For instance, in the elastic strip framework [2] the planner is allowed to temporarily suspend the internal forces keeping two waypoints together and then, when the obstacle is passed, restore them to rejoin the two trajectory branches. However, this formulation is limited to paths defined as a sequences of configurations and it does not guarantee that the two disjoint branches would actually cross to the other side of the obstacle.

Here we propose a procedure that, given an obstacle point $\boldsymbol{o}_i$ on one side of $\boldsymbol{\gamma}(\boldsymbol{x}, s)$, autonomously generates an alternative set of shape parameters $\boldsymbol{x}_{oi} \in \mathbb{R}^n$ such that $\boldsymbol{o}_i$ is on the other side of $\boldsymbol{\gamma}(\boldsymbol{x}_{oi}, s)$. The *alternative path* $\boldsymbol{\gamma}(\boldsymbol{x}_{oi}, s)$ is initialized and generated when $\boldsymbol{o}_i$ induces a big enough deformation on $\boldsymbol{\gamma}(\boldsymbol{x}, s)$. Introducing a threshold $F > 0$, this

[1] The ability to track a sufficiently smooth trajectory is a common property of basically all mobile robots within the scope of this work. This property holds, among the others for all those differentially flat systems [10] whose flat output includes a Cartesian point, or, equivalently, possessing a point that can be linearized through a dynamical feedback [11].

[2] We note that this approach could be seen as an extension of the classical Task-Priority framework developed for robot manipulators and recast to our particular needs, see [12] for more details.

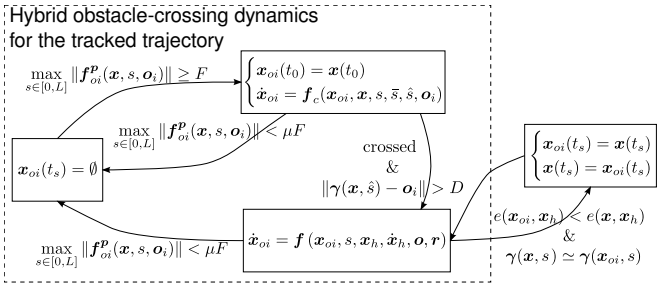


Fig. 2: Block diagram of the hybrid obstacle-crossing dynamics for the tracked trajectory.

condition can be expressed in terms of the repulsive force $\boldsymbol{f}^{\boldsymbol{p}}_{oi}$ in (11) as

$$\max_{s\in[0,L]} \|\boldsymbol{f}^{\boldsymbol{p}}_{oi}(\boldsymbol{x},s,\boldsymbol{o}_i)\| \geq F. \tag{15}$$

Note that, by definition, the maximum value of $\|\boldsymbol{f}^{\boldsymbol{p}}_{oi}\|$ is obtained on the closest point to the obstacle, let this be $\boldsymbol{\gamma}(\boldsymbol{x},\bar{s})$, which can be computed by solving the problem

$$\bar{s} = \min_{s\in[0,L]} \|\boldsymbol{\gamma}(\boldsymbol{x},s)-\boldsymbol{o}_i\|. \tag{16}$$

Assume that condition (15) becomes true at time instant t_0, the alternative path is generated by letting the alternative parameters follow a dynamical system of the form:

$$\begin{cases}\boldsymbol{x}_{oi}(t_0)=\boldsymbol{x}(t_0)\\ \dot{\boldsymbol{x}}_{oi}=\boldsymbol{f}_c(\boldsymbol{x}_{oi},\boldsymbol{x},s,\bar{s},\hat{s},\boldsymbol{o}_i).\end{cases} \tag{17}$$

In order to define $\boldsymbol{f}_c \in \mathbb{R}^n$, we first need to introduce another artificial vector field $\boldsymbol{f}^{\boldsymbol{p}}_c \in \mathbb{R}^d$, i.e., acting in the Cartesian space. The role of $\boldsymbol{f}^{\boldsymbol{p}}_c \in \mathbb{R}^d$ is to apply an action that leads a single given point $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},\hat{s})$ of the alternative path to the other side of the obstacle. Formally, $\boldsymbol{f}^{\boldsymbol{p}}_c$ is designed as

$$\boldsymbol{f}^{\boldsymbol{p}}_c(\boldsymbol{x}_{oi},\boldsymbol{x},s,\bar{s},\hat{s},\boldsymbol{o}_i) = \frac{d\psi(\|\boldsymbol{\gamma}(\boldsymbol{x}_{oi},s)-\boldsymbol{o}_i\|)}{d\|\boldsymbol{\gamma}(\boldsymbol{x}_{oi},s)-\boldsymbol{o}_i\|}\,\boldsymbol{n} \tag{18}$$

where ψ is a strictly increasing artificial potential and $\boldsymbol{n} = \frac{\boldsymbol{o}_i-\boldsymbol{\gamma}(\boldsymbol{x},\bar{s})}{\|\boldsymbol{o}_i-\boldsymbol{\gamma}(\boldsymbol{x},\bar{s})\|}$.

In fact, the point $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},\hat{s})$ where the vector $\boldsymbol{f}^{\boldsymbol{p}}_c$ is applied is given by the intersection of the geometric path $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},s)$ with the line connecting $\boldsymbol{o}_i$ with $\boldsymbol{\gamma}(\boldsymbol{x},\bar{s})$. With the same arguments used before, the desired velocity vector $\boldsymbol{f}^{\boldsymbol{p}}_c$ for the point $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},\hat{s})$ is realized by a velocity vector in the space of $\boldsymbol{x}_{oi}$ using a pseudo-inversion

$$\boldsymbol{f}_c(\boldsymbol{x}_{oi},\boldsymbol{x},s,\bar{s},\hat{s},\boldsymbol{o}_i) = \left.\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{x}}\right|^{\dagger}_{(\boldsymbol{x}_{oi},\hat{s})} \boldsymbol{f}^{\boldsymbol{p}}_c. \tag{19}$$

When the crossing is completed and $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},\hat{s})$ is sufficiently far from the obstacle point $\boldsymbol{o}_i$, then the dynamical evolution of the alternative shape parameters $\boldsymbol{x}_{oi}$ switches to the normal reactive behavior

$$\dot{\boldsymbol{x}}_{oi} = \boldsymbol{f}\left(\boldsymbol{x}_{oi},s,\boldsymbol{x}_h,\dot{\boldsymbol{x}}_h,\boldsymbol{o},\boldsymbol{r}\right). \tag{20}$$

At this point, the alternative collision free path $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},s)$ is fully generated. If at some instant t_s it results that

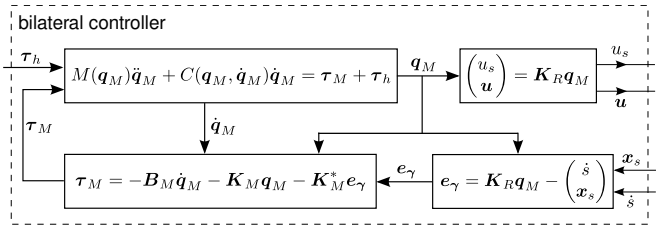


Fig. 3: CM: Block diagram of the bilateral controller.

$e(\boldsymbol{x}_{oi},\boldsymbol{x}_h) < e(\boldsymbol{x},\boldsymbol{x}_h)$, then the alternative path and the tracked one are exchanged, i.e.,

$$\begin{cases}\boldsymbol{x}_{oi}(t_s)=\boldsymbol{x}(t_s)\\ \boldsymbol{x}(t_s)=\boldsymbol{x}_{oi}(t_s).\end{cases} \tag{21}$$

However, the switch is allowed only if $\boldsymbol{\gamma}(\boldsymbol{x},s) \simeq \boldsymbol{\gamma}(\boldsymbol{x}_{oi},s)$ to avoid discontinuities in the tracked trajectory for the robot.

To complete the procedure, if the obstacle $\boldsymbol{o}_i$ gets sufficiently distant from $\boldsymbol{\gamma}(\boldsymbol{x},s)$, i.e. if $\max_{s\in[0,L]} \|\boldsymbol{f}^{\boldsymbol{p}}_{oi}(\boldsymbol{x},s,\boldsymbol{o}_i)\| < \mu F$ with $0 < \mu < 1$, then $\boldsymbol{\gamma}(\boldsymbol{x}_{oi},s)$ is dropped. A block representation of the overall method is depicted in Fig. 2.

The generalization of this procedure to multiple obstacles is straightforward. An alternative path is generated for every obstacle that is applying a strong enough force on the current path, and whenever the current path is switched to another one, all the other alternatives are also reset. This method is not complete at every time instant, however, sequential switches allow to virtually extend the search to large part of the shape-parameter space while still keeping the problem tractable, being the number of path at every instant linear w.r.t. the number of obstacles.

III. Paradigmatic Application to Closed-loop Human-Robot Cooperation

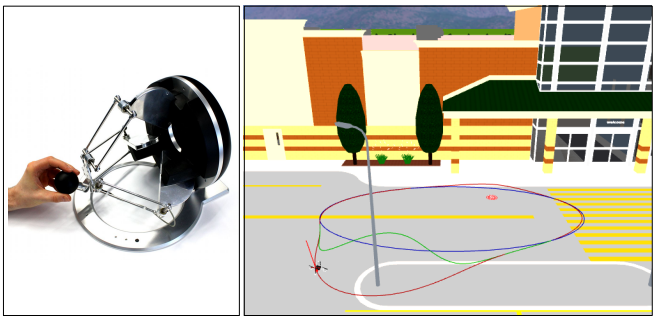
In this section we propose a systematic way to let a human assistant/operator act as MM, thus enabling a fruitful human-robot cooperation. In this case inputs $(u_s,\boldsymbol{u}) \in \mathbb{R}^{m+1}$ are provided by making use of a $(m+1)$-DOF force-feedback device (the master side). The device is modeled as a generic mechanical system whose configuration vector is denoted with $\boldsymbol{q}_M \in \mathbb{R}^{m+1}$, see Fig. 3. The inputs $(u_s,\boldsymbol{u})$ are computed as

$$\begin{pmatrix}u_s\\ \boldsymbol{u}\end{pmatrix} = \boldsymbol{K}_R\boldsymbol{q}_M, \tag{22}$$

where $\boldsymbol{K}_R \in \mathbb{R}^{m+1\times m+1}$ is a positive definite diagonal matrix of scaling factors.

In order to increase the operator situational awareness the force-feedback $\boldsymbol{\tau}_M$ is used to convey information about: i) how well the actual speed $\dot{s}$ is tracking $\dot{s}_h$, i.e., the one commanded by the human via u_s, and ii) how well the whole tracked path $\boldsymbol{\gamma}(\boldsymbol{x})$ is in agreement with the reference path $\boldsymbol{\gamma}(\boldsymbol{x}_h)$ commanded by the human via $\boldsymbol{u}$. Therefore, as a haptic cue we consider the linear combination of two errors in a 'PD-like' fashion:

$$\boldsymbol{e}_\gamma = \boldsymbol{e}_{\dot{\boldsymbol{y}}} + k_y\boldsymbol{e}_y. \tag{23}$$



(a) (b)

Fig. 4: simulation setup for human/hardware in the loop simulations. a): haptic device used to command the task path; b): simulation environment with planned paths.

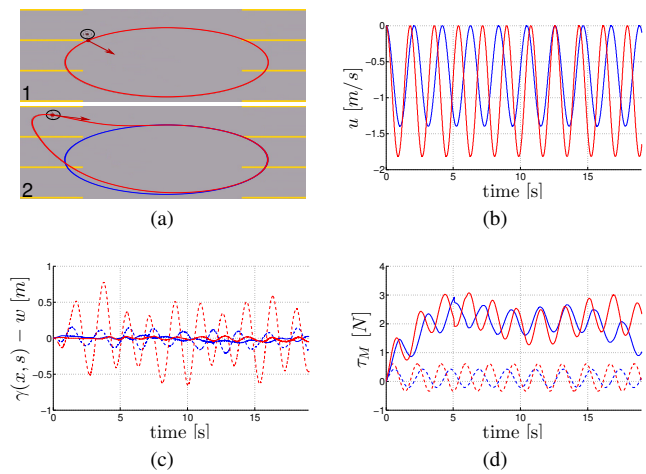


Fig. 5: Simulation comparing 1) the case of no null space projection (solid lines) and 2) when $\dot{\boldsymbol{x}}$ is projected in the null space of $\boldsymbol{J} = \partial\boldsymbol{\gamma}/\partial\boldsymbol{x}|_{(\boldsymbol{x},s)}$ (dashed lines). The red curve in (a) is the tracked trajectory while the blue one is the reference one. The red arrow represents the desired speed. In all the other plots blue and red stand for the x and y component of the signal, respectively.

Here, $\boldsymbol{e}_{\dot{\boldsymbol{y}}}$ represents the 'velocity error' part of (23) and is related to the difference between the commanded $(\dot{s}_h, \dot{\boldsymbol{x}}_h)$ and the executed $(\dot{\boldsymbol{s}}, \dot{\boldsymbol{x}})$, i.e., exploiting (1),

$$\boldsymbol{e}_{\dot{\boldsymbol{y}}} = \begin{pmatrix} \dot{s}_h - \dot{s} \\ \boldsymbol{G}(\boldsymbol{x}_h)\dot{\boldsymbol{x}}_h - \boldsymbol{G}(\boldsymbol{x})\dot{\boldsymbol{x}} \end{pmatrix} = \boldsymbol{K}_R \boldsymbol{q}_M - \begin{pmatrix} \dot{s} \\ \boldsymbol{G}(\boldsymbol{x})\dot{\boldsymbol{x}} \end{pmatrix}. \tag{24}$$

The second term $k_y \boldsymbol{e}_{\boldsymbol{y}}$ in (23) represents the 'position error' term and is associated to the mismatch between the desired shape $\boldsymbol{x}_h$ and its actual implementation $\boldsymbol{x}$, i.e.,

$$\boldsymbol{e}_{\boldsymbol{y}} = \begin{pmatrix} 0 \\ \boldsymbol{G}(\boldsymbol{x})\,(\boldsymbol{x}_h - \boldsymbol{x}) \end{pmatrix}. \tag{25}$$

The use of a force feedback based on the entire planned motion is a new feature of our approach w.r.t. to the more classical haptic-cue algorithms where only the local mismatch between the current robot velocity and the commanded one is used, instead.

The master control is then implemented as

$$\boldsymbol{\tau}_M = -\boldsymbol{B}_M \dot{\boldsymbol{q}}_M - \boldsymbol{K}_M \boldsymbol{q}_M - \boldsymbol{K}_M^* \boldsymbol{e}_\gamma \tag{26}$$

where $\boldsymbol{B}_M$ is a positive definite damping matrix used to stabilize the device, $\boldsymbol{K}_M$ is a diagonal non-negative matrix used to provide a perception of the distance from the zero-commanded velocity, and $\boldsymbol{K}_M^*$ a diagonal positive definite matrix of gains. The resulting scheme is depicted in Fig. 3.

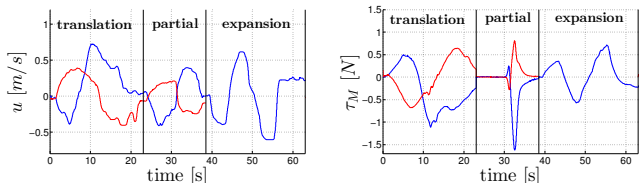
As in all bilateral teleoperation applications, presence of the force feedback $\boldsymbol{\tau}_M$ may cause unstable behaviors of the haptic interface because of non-modeled dynamics, communication delays and packet losses, etc. In order to guarantee stability despite all these shortcomings, we make use of the *passive set-position modulation* (PSPM) approach, a very general and flexible framework for guaranteeing stability (passivity) of the master side and of the closed-loop system [13].

A. Human/Hardware-in-the-loop Simulations

We performed several hardware/human in-the-loop simulations in order to test the proposed framework and its application to the bilateral human/robot cooperation case. The considered physically-simulated robot is a standard quadrotor UAV (see Fig. 4a), the path $\boldsymbol{\gamma}(\boldsymbol{x}, s)$ is parametrized as a fifth-order planar B-spline. We also asked the TS to produce a closed initial path, as described in [9], in order to show the applicability of the proposed framework to typical monitoring/surveillance scenarios where a repetitive motion is often required. The commands $(u_s, \boldsymbol{u})$ are provided by the assistant through an Omega.6 haptic device (Fig. 4b) with 3 actuated degrees of freedom. A video of the simulations is attached.

In the first simulation we evaluate the effects introduced by the null space projector in (8) to keep the current position of the tracked trajectory invariant to $\dot{\boldsymbol{x}}$. The MM applies a planar sinusoidal translation to the reference trajectory (Fig. 5b) in an obstacle-free environment. The case with the null-space projector (Fig. 5a-2) is compared against the case without it (Fig. 5a-1). Figure 5c shows the tracking error $\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{w}$, where $\boldsymbol{w} \in \mathbb{R}^2$ is the planar position of the UAV. When the null space projection is applied (solid lines) the tracking error is visibly smaller than when no projection is used (dashed lines). This confirms the beneficial effects of keeping the local geometric properties of the curve in the point to be tracked. On the other hand, the deformation introduced by the projector $\boldsymbol{N}$ (see Fig. 5a-2) causes a mismatch between $\boldsymbol{x}$ and the reference $\boldsymbol{x}_h$, even in free space. This effect is reflected on the force feedback $\boldsymbol{\tau}_M$ (Fig. 5d), that becomes informative of the inertia of the path to these local changes.

The second simulation offers an example of how the MM can change the reference trajectory using different maps $\boldsymbol{G}(\boldsymbol{x_h})$ such as global-translation local-translation and expansion. The MM commands are plotted in Fig. 6a, where vertical black lines indicate the change to a different command map, while the force feedback $\boldsymbol{\tau}_M$ is depicted in Fig. 6b. Notice how the force feedback (Fig. 6b) when using the partial translation map (from $t = 25$ s to $t = 38$ s) is not null only for a short period. This is due to the fact that the map produces a local modification of the path, therefore the



(a) (b)

Fig. 6: Second simulation showing how the changes in the reference trajectory (blue curve) affect the tracked trajectory (red curve), and generates a haptic feedback to the human assistant.

null space projection produces a mismatch only when the robot is traveling on this part.

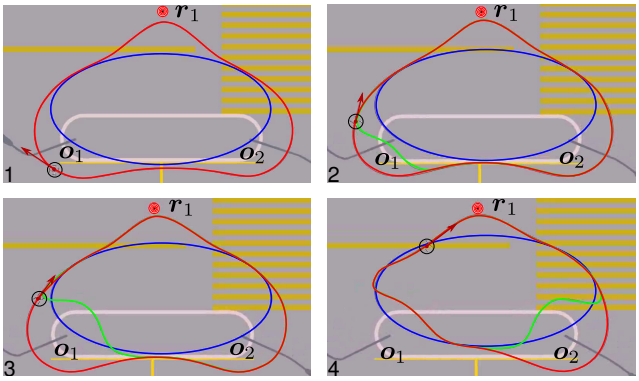
Finally, the third simulation demonstrates the behavior of the obstacle-crossing dynamics for the tracked trajectory described in Sec. II-B. Figure 7a shows four crucial moments of the simulation: 1) the tracked trajectory is deformed by the two obstacles, $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$, and attracted by the target $\boldsymbol{r}_1$; 2) condition (15) is met for $\boldsymbol{o}_1$ and the alternative path (green line) is being generated according to (19); 3) the alternative path reaches a better deformation according to (17), however the RM cannot switch to it because $\boldsymbol{\gamma}(\boldsymbol{x}, s) \neq \boldsymbol{\gamma}(\boldsymbol{x}_{o1}, s)$; 4) the RM has now switched to the alternative path, and another alternative has been generated by $\boldsymbol{o}_2$. All the switches in the replanning are denoted by solid vertical black lines in the plots of Figs. 7b-7c, while a dashed vertical line indicates that the point of interest is detected. Figure 7b shows the evolution of the average error $e(\boldsymbol{x}, \boldsymbol{x}_h) = \sum_{i=1}^{n} |\boldsymbol{x}_{h,i} - \boldsymbol{x}_i||$. After every path switch the error becomes smaller, confirming the usefulness of the proposed crossing procedure.

Finally, Fig. 7c shows the evolution of the force feedback $\boldsymbol{\tau}_M$. As expected, the force feedback becomes stronger when the tracked trajectory is deformed by the obstacle. Note also that the discontinuities in the feedback when a switch occurs is helpful to inform the human that tracked trajectory has crossed an obstacle and can therefore continue more easily. Similarly, the sudden force that is felt when a point of interest is within range naturally guides the human operator in directing the the reference trajectory towards it.

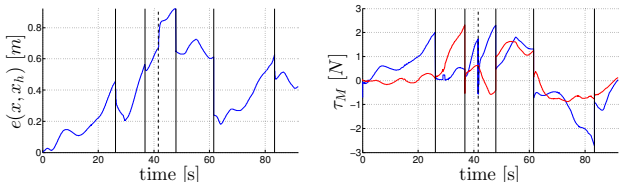
IV. Conclusions

In this work we have presented a new framework for the synergetic combination of an task scheduler, a cognitive-based planner (e.g., sophisticated algorithm or a human assistant), and a reactive (low-level) planner. We also described the design of a bilateral (haptic) connection between the human assistant and the reactive planner which benefits from a novel idea based on the deformation of the whole path. Effectiveness of the proposed approach has been demonstrated through human/hardware in-the-loop physical simulations.

Future developments include the implementation of this framework with a real mobile robot (e.g. a UAV) the extension to a multi-robot scenario.



(a)



(b) (c)

Fig. 7: Third simulation showing the action of the obstacle-crossing dynamics that leads the tracked trajectory (red curve) to switch to an alternative path (green) in order to better match the reference path (blue).

References

[1] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *1993 IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, May 1993, pp. 802–807.

[2] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[3] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Trans. on Robotics*, vol. 20, no. 6, pp. 967–977, 2004.

[4] N. Diolaiti and C. Melchiorri, "Teleoperation of a mobile robot through haptic feedback," in *IEEE Int. Work. on Haptic Virtual Environments and Their Applications*, Ottawa, Canada, Nov. 2002, pp. 62–72.

[5] D. J. Lee, O. Martinez-Palafox, and M. W. Spong, "Bilateral teleoperation of a wheeled mobile robot over delayed communication network," in *2006 IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, May 2006, pp. 3298–3303.

[6] A. Franchi, P. Robuffo Giordano, C. Secchi, H. I. Son, and H. H. Bülthoff, "A passivity-based decentralized approach for the bilateral teleoperation of a group of UAVs with switching topology," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 898–905.

[7] A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of groups of mobile robots with time-varying topology," *IEEE Trans. on Robotics*, In Press, Electronically published at http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6199993.

[8] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs." *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, 2012.

[9] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Interactive planning of persistent trajectories for human-assisted navigation of mobile robots," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012.

[10] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[11] A. Isidori, *Nonlinear Control Systems, 3rd edition*. Springer, 1995.

[12] B. Siciliano and O. Khatib, *Handbook of Robotics*. Springer, 2008.

[13] D. J. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.